*Journal of Computer and Knowledge Engineering, Vol. 6, No. 2, 2023. (41-56)*

Ferdowsi University of Mashhad

# Journal of Computer and Knowledge Engineering

https://cke.um.ac.ir

Information and Communication Technology Association of Iran

# Dynamic Security Risk Management Considering Systems Structural and Probabilistic Attributes*

Research Article

Masoud Khosravi-Farmad[1], Abbas Ghaemi-Bafghi [2]

*DOI: 10.22067/cke.2023.83744.1102*

**Abstract:** Today's cyber-attacks are getting more sophisticated and their volume is consistently growing. Organizations suffer from various attacks in their lifetime each of which exploiting different vulnerabilities, therefore, preventing them all is not affordable nor effective. Hence, selecting the optimal set of security countermeasures to protect IT assets from being compromised is a challenging task which requires various considerations such as vulnerabilities characteristics, countermeasures effectiveness, existing security policies and budget limitations. In this paper, a dynamic security risk management framework is presented which identifies the optimal risk mitigation plans for preventing ongoing cyber-attacks regarding limited budget. Structural and probabilistic analysis of system model are conducted in two parallel and independent aspects in which the most probable system's risk hotspots are identified. Suitability of countermeasures are also calculated based on their ability in covering vulnerabilities and organizational security policies. Moreover, a novel algorithm for dynamically conducting cost-benefit analysis is proposed which identifies optimal security risk mitigation plans. Finally, practical applicability is ensured by using a case study.

**Keyword:** Attack Graph, Bayesian Networks, Cost-Benefit Analysis, Countermeasure Analysis, Security Risk Management.

## 1. Introduction

By evolving technology and increasing the number of sophisticated IT related threats and growth of cybercriminals capabilities in exploiting security vulnerabilities, reducing security risks by protecting valuable assets is becoming the greatest concern for digitized companies. Risk is defined as the net negative impact of the exploitation of security vulnerabilities and is determined by considering the probability of successful exploit of vulnerabilities and the impact they incur on the confidentiality (C), integrity (I), and availability (A) requirements of assets, known as CIA requirements [1]. Security risk management generally is the process of identifying, assessing and mitigating risks to an organization's IT assets [2, 3]. The risk identification stage is the process of identifying assets and their significance in bringing an organization closer to its goals. Vulnerabilities putting the CIA requirements of these assets at risk are also identified in this stage. The risks are then determined using probability of exploiting assets vulnerabilities combined with their overall consequences in the risk assessment stage. After that, in the risk mitigation stage, the highest-ranked risks are selected to be treated by lessening the probability and/or impact of them.

A variety of security risk assessment methodologies have been proposed in the literature which can be broadly categorized into qualitative, quantitative and semi-qualitative methods [4, 5]:

1. In qualitative methods such as [2, 6], Information security risks are assessed using relative non numerical values (e.g. low, medium, and high). These methods are useful for dealing with situations which are not well defined. While qualitative methods are simple and easy to understand and implement, they lack enough accuracy and precision in calculations involved. Also, these methods are based on the knowledge and experience of assessors, making them more subjective and error prone than quantitative methods [2]. Moreover, since the range of qualitative values are relatively small, risk prioritization and comparison is comparatively difficult [1].
2. In quantitative methods such as [7-9], Information security risks are assessed using numbers. These methods are based on objective measurements, hence, the results are more accurate and clear. While these methods have advantages, they meet several problems. For instance, because of limited time, budget, and human resources available, their implementation complexity is more than their qualitative counterparts. Moreover, exact detailed information about system attributes may not always be easily extractable from experts when not enough accurate historical data is

* Manuscript received: 2023 August 12, Revised, 2023 September 9, Accepted, 2023 November 1.

[1] Corresponding author. Associate Professor, Data and Communication Security Lab., Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran. **Email**: m.khosravi@mail.um.ac.ir.

[2] PhD Student, Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran.

available [10].

3. Semi-qualitative methods such as [11, 12], try to combine advantages of both quantitative and qualitative methods. They can benefit the simplicity and understandability of the qualitative methods, while taking advantage of the accuracy of quantitative methods.

There are several standards available for assessing individual IT systems security vulnerabilities. One of the most common and widely adopted standards is Common Vulnerability Scoring System (CVSS) [13] developed by the Forum of Incident Response and Security Teams (FIRST) [14] which assigns both numerical scores and relative values to identified vulnerabilities.

Most of existing approaches take into account the overall scores of vulnerabilities for assessing systems security risks. These values usually represent the security level of coarse-grained attributes of IT systems. This viewpoint makes the process of risk assessment done straightforward, but it faces obstacles in risk mitigation process when cost-benefit analysis is required to determine appropriate risk control recommendations [5]. For instance, consider three vulnerabilities existing on .NET Framework developed by Microsoft which are listed in Table 1. Suppose that, in case of budget limitation, the security administrator tries to specify a proper countermeasure for covering only one of these vulnerabilities regarding their scores. As can be seen, these vulnerabilities are hard to be distinguished based on only their CVSS Base Scores. Because, despite the major difference, all of them have the same score equal to 7.5. To overcome this problem, the values of CIA parameters of vulnerabilities should also be taken into account in decision making. For example, if confidentiality is more important for an asset, then the security administrator should identify a countermeasure covering CVE-2016-0047 and leave other vulnerabilities unpatched. For more information about CVSS scoring system refer to [13].

Since during modern sophisticated attacks such as Advanced Persistent Threats [15-17] sequences of vulnerabilities are usually exploited to perform multi-step attacks to achieve particular goals, utilizing only individual scores or values of these standards is not sufficient, because they do not consider the interactions between vulnerabilities. In order to be able to clearly demonstrate multi-step attacks, one can use graph-based security models such as Attack Graph (AG) [18-20]. AG is a powerful model that can encode causal relationships between vulnerabilities and give description about the correlated attacks.

Risk mitigation is a crucial stage in the process of risk management which is required for successfully reducing systems security risks [2]. It includes prioritizing, implementing and maintaining the most suitable security countermeasures. The input data for risk mitigation is provided from risk assessment results. Therefore, a risk assessment report is beneficial only when it is compatible with risk mitigation processes.

In this paper, we present a method for managing IT systems security risks which uses both numerical and relative values. In cases when input values are supplied by experts and security administrators, relative values are used to ease the process of data extraction. In cases when enough data is available in existing security databases or repositories, exact values are used. We use AG as a graphical security model for modeling different attack scenarios targeting IT assets. We analyze AGs in two ways:

1. **Structural analysis**: Each AG contains several attack paths each of which represents an attack scenario. Therefore, it is an important source for extracting attackers' behavioral information to identify existing risk hotspots. We identify these risk hotspots using defined metrics over the structure of AG.
2. **Probabilistic analysis**: by assigning a probability to each node of AG and applying Bayesian theory we can compute unconditional probability (UP) of attackers reaching to different states in the graph. These probabilities will play an important role in final risk reducing decision making.

Moreover, a parametric solution for countermeasure analysis is presented which first calculates the coverage level of vulnerabilities by countermeasures based on fine-grained attributes. After that, considering organizational security policies on assets, the suitability of countermeasures for implementation is identified.

Briefly, the main contributions of this work are:

1. A dynamic security risk management framework is presented which uses exact values when enough data is available and uses relative values when data need to be extracted from experts' knowledge.
2. Structural and probabilistic analysis of AG model are conducted in two parallel ways. In structural analysis, AGs risk hotspots are identified and in probabilistic analysis the UP of attackers reaching their goals are calculated utilizing Bayesian theory.
3. The countermeasures' ability in reducing vulnerabilities impact are calculated in terms of C, I and A parameters.
4. Utility of countermeasures are calculated based on their ability in reducing vulnerabilities, their negative effects on the service level agreements, organizational security policies and systems risk hotspots.
5. A novel algorithm for dynamically conducting cost-benefit analysis is presented which identifies optimal security risk mitigation plans.

The paper is structured as follows. Section 2 reviews related work. Section 3 presents the concepts used for modeling systems attributes. The proposed risk management framework is presented in Section 4. Experimental results are given in Section 5. Finally, Section 6 concludes the paper and discusses future work.

Table 1. Sample .Net Framework Vulnerabilities

| CVE ID | Confidentiality (C) | Integrity (I) | Availability (A) | CVSS v3 Base Score | Vector |
|---|---|---|---|---|---|
| CVE-2016-0047 | High | None | None | 7.5 (High) | CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N |
| CVE-2017-0248 | None | High | None | 7.5 (High) | CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N |
| CVE-2016-0033 | None | None | High | 7.5 (High) | CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H |

## 2. Related Work

A variety of cybersecurity risk management methodologies have been developed for assessing risks in IT systems, thereby, enabling systems security administrators to make correct decisions towards mitigating the most important risks in the operational environments.

Qualitative security risk management methods rate both input and output attributes of a system using a scale of usually three or five levels (e.g., very low, low, moderate, high, very high) [2, 21]. Since these methods have several disadvantages, including their inappropriateness in making a cost-benefit analysis of recommended controls, quantitative methods are preferred. The major advantage of a quantitative method is that it most effectively supports cost-benefit analysis of alternative risk-reducing measures [1]. Quantitative methods such as [9, 22-25] typically employ sets of methods, principles or rules for managing risks based on the use of numbers. Semi-qualitative methods can provide the benefits of quantitative and qualitative methods [1].

Because AG-based security models can properly model multi-step attacks, they are popular in both qualitative and quantitative risk management activities [18, 19, 26-31]. Some approaches apply Bayesian concept over AG to represent information about causal relationships between vulnerabilities and capture uncertainties about probabilities of attacker actions. One of the first researches in this field is [32] which models attack paths using Bayesian networks and quantitatively represents the security of computer networks. Frigault and Wang [33] used Bayesian networks with AGs to calculate security metrics. They named their model Bayesian attack graph. After that, Poolsappasit et al. [34] extended their model to be able to dynamically analyze existing risks in networked systems. In [35], a security risk analysis model based on Bayesian networks and ant colony optimization algorithm is proposed which estimates risk values. In [9, 36] authors used Bayesian networks to implement Factor Analysis of Information Risk (FAIR) as one of the most popular models for quantitative security risk assessment. There are several works that use Bayesian networks for assessing security risks of IT systems and capturing uncertainties in attacker actions, such as [37-42]. Aforementioned methodologies use Bayesian inference results for calculating the risk level of systems and do not consider anatomy of attack scenarios and their interactions in forming successful attacks. The methodology proposed in this paper not only uses Bayesian inference results, but also takes into account the topology and structure of security model in risk assessment calculations.

While most of existing researches focus on risk assessment and vulnerability analysis, fewer studies proposed methods for risk mitigation and countermeasure analysis. The reason is that there is no standard and comprehensive database for security countermeasures [43]. Moreover, most of risk mitigation processes are largely dependent to expert's knowledge. In [44], minimum-cost countermeasures are identified using exploit dependency graphs. In [45], the minimal subset of attacks that are necessary for reaching a goal in the network is determined. After that, the minimal subset of countermeasures that covers the subset of attacks is identified. Dewri et al. [46] used a multi-objective optimization problem on the security model of the network to determine if a given set of security hardening measures effectively secures the system. In [34], authors proposed a Bayesian attack graph model which assigns cost and outcome values to each countermeasure. After that, by applying genetic algorithm solutions, countermeasures with the highest outcome given a specific budget are identified. Authors in [47] proposed Bayesian decision networks to manage security vulnerabilities and conduct cost-benefit analysis by using a variable elimination-based algorithm to identify the optimal subset(s) of security countermeasures. Authors in [48] assign an effectiveness value to each countermeasure. This value represents the percentage of probability reduction of vulnerabilities for which the countermeasure is implemented on. In [49], authors define safeguard effectiveness as the ability of safeguards in reducing the criticality of threats. Authors in [50] define countermeasure effectiveness as risk mitigation level after the countermeasure implementation. The mentioned methods are useful, but the main problem with such methods is that countermeasures effectiveness is assigned statically and security experts are responsible to assign numeric outcomes to each countermeasure solely, regardless of systems vulnerabilities and ongoing attacks. This, indeed, is not an easy task, because, outcome of a countermeasure is dependent on its ability in remedying its covered vulnerabilities and dynamic state of the system. Moreover, most of existing researches neglect countermeasures' negative impact on service quality and service level agreements which may lead to select inappropriate countermeasures and as a result, reducing network performance.

In this paper, a dynamic security risk management framework is presented which utilizes fine-grained attributes of IT systems to handle the aforementioned drawbacks in

existing methods. Using relationships between security policies on assets, security vulnerabilities existing on assets and countermeasures covering these vulnerabilities, we can manage IT systems security risks properly.

### 3. Modeling system attributes

Application of the proposed framework requires a keen understanding of the system-related information. Hence, we need to model the attributes of the system under assessment appropriately. For this reason, we define security attributes as necessary system-related information sources for doing risk management. We categorize the security attributes into three main classes, namely, Assets, Vulnerabilities and Security Countermeasures. Each attribute has three requirements, namely, confidentiality (C), integrity (I) and availability (A), known as CIA requirements. Asset's CIA represent the importance of CIA requirements of assets which are determined according to the organizations security policies. Each vulnerability can impact assets in terms of CIA parameters. Security countermeasures can reduce this impact and therefore protect assets from being compromised. The relations between the security attributes is depicted in Figure 1.

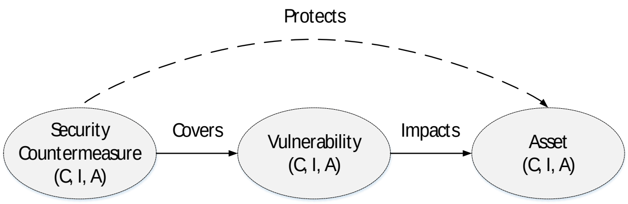



Figure 1. Relations between Assets, Vulnerabilities and Security Countermeasures

The information for each of the attributes are modeled using a vector. A brief description about each vector and its components is presented below:

- **Asset (A)**: we define assets as any hardware or software component in the system under assessment which supports information-related activities. Therefore, assets could be hosts existing in the system, operating systems, services and software running on them. The CIA requirements of assets are also included in vector. Moreover, each asset is assigned a unique identifier. Hence, the Asset Vector is defined as bellow:

Asset = {Identifier, Asset Details (Name, Operating System, Service, Software, Hardware, Protocol, etc.), CIA Requirements}

- **Vulnerability (V)**: we define vulnerability as any weakness or flaw existing on an asset configuration which could be exploited by malicious attackers and result in violation of the system’s CIA requirements. Each vulnerability is associated with a CVE ID which is a unique identifier for publicly disclosed information security vulnerabilities. The impact on CIA requirements of successful exploitation of vulnerabilities and the exploitation probability of vulnerabilities are also included in the vector. Therefore, the Vulnerability Vector is defined as bellow:

Vulnerability={$CVE_{ID}$, Vulnerable Asset Configurations (Operating System, Service, Software, Hardware, Protocol, etc.), Impact on CIA, Exploitation Probability}

- **Security Countermeasure (SC)**: a security countermeasure or a security control is a protecting measure which reduces the vulnerability of an asset by protecting its CIA requirements. Each SC is assigned a unique identifier and also the IDs of covered vulnerabilities. We separate the efficacy of SCs into two classes; 1- a SC could reduce the impact of a vulnerability on CIA requirements of an asset (impact reduction (IR)), and/or 2- it could reduce the exploitation probability of a vulnerability (probability reduction (PR)). Implementation of SCs may bring negative effects on the service level agreements. Therefore, we define intrusiveness (I) which reflects this effect. Moreover, each SC has a cost of implementation (IC). Hence, the Security Countermeasure Vector is defined as bellow:

Security Countermeasure = {Identifier, IDs of Covered Vulnerabilities, Impact Reduction on CIA (IR), Probability Reduction (PR), Intrusiveness (I), Implementation Cost (IC)}

In the next Section, we express how these vectors are used in the proposed security risk management framework.

### 4. The Proposed Risk Management Framework

The proposed dynamic security risk management framework, consists of four activities, namely, vulnerability scanning, modeling network attacks, countermeasure analysis and dynamic cost-benefit analysis. These activities contain seven processes, namely, vulnerability scanning, attack graph generation, attack graph analysis, Bayesian inference, vulnerability coverage assessment, policy conformance assessment and cost-benefit analysis. It starts by identifying the vulnerabilities existing on system assets. After that, there are two activities which could be done in parallel. The first activity is modeling network attacks, in which the AG model is used to model attack scenarios. The generated model is then analyzed and some structural metrics are extracted from it. These metrics will be used in dynamic cost-benefit analysis process. Moreover, to consider the interconnections between vulnerabilities, Bayesian inference algorithm is applied on the AG model. Therefore, UP of compromising each network state is calculated. These probabilities are also used in dynamic cost-benefit analysis process. In the second activity, i.e. countermeasure analysis, a mapping analysis is conducted to calculate the coverage level of vulnerabilities impact by SCs in terms of C, I and A parameters. After that, the conformance between coverage levels of vulnerabilities with the importance of C, I and A parameters of assets are calculated regarding organizational security policies. Therefore, the suitability of countermeasures for covering vulnerabilities existing on specific assets can be calculated. Finally, considering extracted metrics from AG model, UP of network states and suitability of each SC, a cost-benefit analysis is conducted to identify the optimal security risk mitigation plans to reduce the overall risk level of the system. The data flow diagram of the proposed dynamic security risk management framework is depicted in Figure 2.

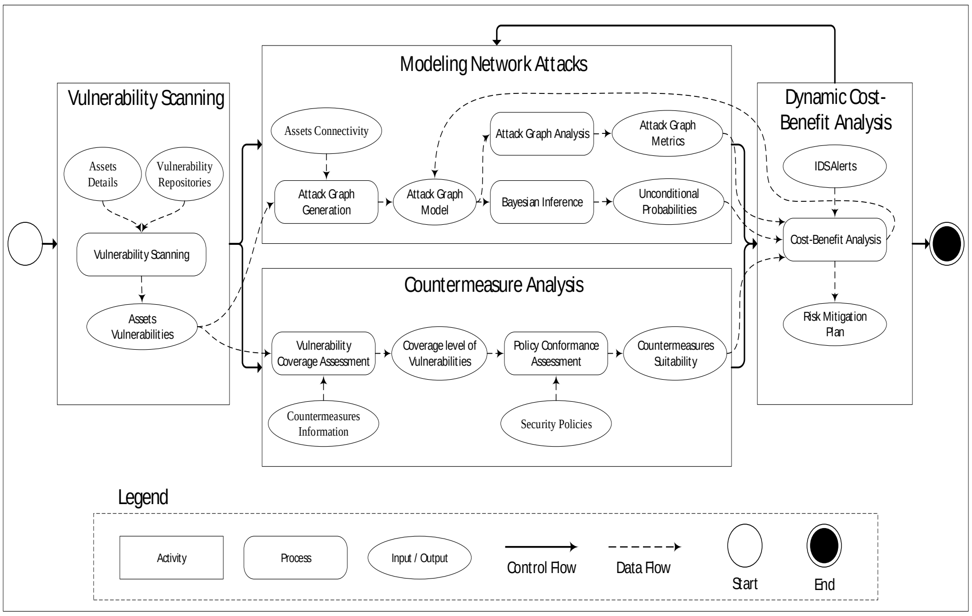



Figure 2. The Proposed Security Risk Management Framework

In this section, each of the mentioned activities and processes of the proposed risk management framework is described in detail.

### *4.1. Vulnerability Scanning*

In this activity, all of the system's assets such as hardware, software, operating systems and services are scanned to search for security vulnerabilities. There exist several vulnerability scanners that can be used for this purpose such as Nessus [51], OpenVAS [52] and Retina [53]. After finding a vulnerability, the information about it can be extracted from existing vulnerability repositories such as the US National Vulnerability Database (NVD) [54] and MITRE's Common Vulnerabilities and Exposures (CVE) [55]. The vulnerabilities information containing their technical description, CVE ID and metrics values gathered in this activity will be used in the next steps. After discovering the system's assets vulnerabilities, we define an Asset Vector for each vulnerability. Moreover, vulnerabilities existing on each asset are listed in a table called Assets Vulnerabilities for future use. This table represents a mapping between assets and vulnerabilities existing on them.

### *4.2. Modeling Network Attacks*

This activity consists of three processes, namely, attack graph generation, attack graph analysis and Bayesian inference. At first, the AG model of the system under assessment is generated. After that, the generated model is processed in two parallel processes. In the attack graph analysis process, some structural metrics of the graph are extracted which represent the risk hotspots in the graph. In the Bayesian inference process, the AG model is converted into the Bayesian attack graph. Therefore, by applying Bayesian theorem, the UP of each network state being compromised by attackers is calculated.

The three mentioned processes are further explained in the following subsections.

### *A. Attack Graph Generation*

This process aims at generating a model based on the information about assets vulnerabilities and their connectivity. Here, the assets connectivity is an important factor in modeling process, because, most of modern sophisticated attacks utilize several vulnerabilities existing on different assets in various sequences. These attacks are called multi-step attacks which can be properly represented by attack graph model. Attack graph is a powerful tool that can demonstrate all attack scenarios an adversary can utilize to compromise a system by modeling vulnerabilities and interactions between them. Therefore, each node of the attack graph represents a state in which a vulnerability exploits.

**Attack Graph Definition**: An attack graph (AG) is a tuple $AG = (S, s_0, s_g, \tau)$, where:

- $S$ is a set of states in the network. Each state represents an exploitation of a vulnerability.
- $s_0 \subseteq S$ denotes the attacker's entry point to the network and hence is the initial state in the graph.
- $s_g \subseteq S$ is the set of potential goals for attackers.
- $\tau \subseteq S \times S$ is the set of directed arcs that change the states of the network.

To generate an AG for a given system, information about vulnerabilities existing on assets and connectivity of assets are required. Information about assets vulnerabilities are provided from output of Vulnerability Scanning process. To determine how assets are connected together, one can refer to the documentations about system topology provided by system administrator or use available tools like Nmap [56] security scanner. After identifying assets vulnerabilities and their connectivity, AG model of the system can be generated using existing tools such as MulVAL [57] and TVA [58].

### *B. Attack Graph Analysis*

After generating AG model of the system under assessment, its risk hotspots should be identified. We define risk hotspots as the most important nodes within the structure of an AG.

To find these hotspots, four AG structural metrics, namely exposure, path length, closeness centrality and betweenness centrality are used in this paper which are defined as follows:

**Exposure** for $node_i$ is defined as the summation of indegree and outdegree of $node_i$ in the AG model.

**Path length** for $node_i$ is defined as the length of the shortest path between the leaf node (i.e. attacker's entry point) and the root node of the AG model (i.e. attacker's goal) visiting $node_i$.

**Closeness centrality** for $node_i$ is defined as the length of the shortest path between $node_i$ and the root node of the AG model. It represents how close $node_i$ is to attacker's goal.

**Betweenness centrality** for $node_i$ is defined as the number of paths that pass between the leaf node and the root node of the AG model visiting $node_i$.

Using the defined metrics, we present Equation 1 to identify the importance level, i.e. centrality, of each state node in the attack graph:

$$importance = \left(\frac{exposure \times betweenness\ centrality}{path\ length \times closeness\ centrality}\right) \quad (1)$$

As can be seen, in Equation 1, importance of a node have a direct relation with its exposure and betweenness centrality, but have an indirect relation with its path length and closeness centrality. In fact, if a node gets involved in more attack scenarios and if these scenarios are more critical, then this node is at higher importance and represents a more at-risk spot in the system.

### *C. Bayesian Inference*

A Bayesian network is a probabilistic graphical model which uses Bayesian inference techniques for probability computations [59]. In order to be able to apply Bayesian inference over the generated AG model, conditional probability tables should be added to each of its nodes [8, 60]. Each conditional probability table represents conditional probability of a network state with respect to its parents. The entries of conditional probability tables are filled with the probabilities of vulnerabilities exploitations. To calculate the exploitation probability of vulnerabilities, their CVSS Base scores are used in this paper. CVSS is an open framework which provides a way to assess the severity level of IT vulnerabilities [13]. Since the CVSS's scores are in the interval of [0 – 10], we divide them by 10. As the result, it is possible to calculate the UP of compromising network states by attackers [60]. UP of a network state indicates the likelihood that this state gets compromised independent of whether any other states are compromised by attackers. States with higher unconditional probability represent gears that attackers can easily take advantage of. More detailed information about Bayesian inference techniques and algorithms can be found in [59, 61].

## *4.3. Countermeasure Analysis*

This activity consists of two processes, namely, vulnerability coverage assessment and policy conformance assessment which are explained in the following subsections.

### *A. Vulnerability Coverage Assessment*

Before conducting vulnerability coverage assessment, we need to find SCs covering system vulnerabilities and define a Security Countermeasure Vector (see Section 3) for each one of them. This information can be gathered from vulnerability repositories, publicly available security reports and documents and also security administrator knowledge.

After defining Security Countermeasures Vectors, we need to assess the efficacy of SCs coverage, i.e. impact reducing ability, over their covered vulnerabilities. For this reason, we propose a table called Coverage Table for each one of the C, I and A requirements separately which is shown in Table 2.

Table 2. Coverage Table Definition for C/I/A Requirement

| Vulnerability (C/I/A) | Security Countermeasure (C/I/A) | Coverage Level |
|---|---|---|
| None | None | Equal coverage |
| None | Partial | Extra coverage |
| None | Complete | Extra coverage |
| Partial | None | No coverage |
| Partial | Partial | Equal coverage |
| Partial | Complete | Extra coverage |
| Complete | None | No coverage |
| Complete | Partial | Little coverage |
| Complete | Complete | Equal coverage |

The first column of this table represents the magnitude of impact on C/I/A caused by exploitation of a vulnerability. The second column represents the ability of a SC to mitigate the C/I/A impact of its covered vulnerabilities. And the third column represents the coverage level resulted by implementing a SC on its corresponding vulnerability in terms of C/I/A.

For instance, the first row of this table can be interpreted as follow: If exploitation of vulnerability $V_i$ doesn't have any impact on C/I/A requirement of an asset, then existing a security countermeasure $SC_i$ with no coverage on C/I/A requirement (or even absence of security countermeasure) results in equal coverage. As another example, the second row of the table says that, if exploitation of vulnerability $V_i$ doesn't have any impact on C/I/A requirement of an asset, then existing a security countermeasure $SC_i$ with partial coverage on C/I/A requirement results in extra coverage. The same interpretations go for the rest of the table.

The output of this process is the C/I/A coverage level for each pair of $< Vulnerability, Security\ Control >$.

### *B. Policy Conformance Assessment*

After conducting vulnerability coverage assessment, the coverage levels of vulnerabilities by their covering SCs are available. But this criterion is not sufficient for selecting appropriate SCs, because it doesn't reflect importance of security policies over the CIA requirements of the organizational assets. Hence, we need to consider assets CIA requirements importance in decision making process. To do so, we propose a table called Suitability Table for each one of the C, I and A requirements separately which is shown in Table 3. SCs with higher suitability are more appropriate to be selected to mitigate the risks.

Table 3. Suitability Table Definition for C/I/A Requirement

| Coverage Level | Assets C/I/A Requirement Importance: None | Partial | Complete |
|---|---|---|---|
| No coverage | 5 | 3 | 1 |
| Little coverage | 4 | 4 | 2 |
| Equal coverage | 3 | 7 | 9 |
| Extra coverage | 2 | 6 | 8 |

The rows of Table 3 represent the coverage levels of SCs over their covered vulnerabilities acquired from Table 2. The columns of this table represent the importance of security policies over the C/I/A requirement of assets. The values of table's entries represent relative suitability of countermeasures implementation based on their ability in covering vulnerabilities and organizational security policies.

For example, if an asset's C/I/A requirement importance is ranked as Complete, then leaving it unprotected, i.e. providing No Coverage, results Suitability of 1 (lowest suitability). But, if we provide Equal Coverage for this asset, we gain the Suitability of 9 (highest suitability).

### *4.4. Dynamic Cost-Benefit Analysis*

During the lifetime of IT systems, they could be targeted to many attacks. It's important to dynamically respond to these attacks to stop attackers' progress and their further intrusions. In this activity, we conduct a dynamic cost-benefit analysis to find the optimal security risk mitigation plans (SRMPs). A SRMP is a subset of SCs which are selected for implementation to cover the most important vulnerabilities and mitigate the overall security risk level of a system. As the result of this step, SRMP with the highest utility and total implementation cost lower than the allocated security hardening budget is identified.

Algorithm 1 represents the steps for dynamically identifying the optimal SRMP for a given attack scenario. The inputs to the algorithm are generated alert by IDS, representing a network state is compromised, AG model of the system under assessment, SC Vectors, SC suitability tables, importance level of AG nodes and allocated budget for system hardening. The output of the algorithm is the optimal SRMP considering budget limitation.

The algorithm starts by selecting the corresponding state to the generated IDS alert, i.e. $state_{Compromised}$ (line 1). Since an alert is generated only when the attack has taken place, we set the probability of $state_{Compromised}$ to 1 and after that, new probabilities of other states in the model are recalculated by applying Bayesian inference over the AG model (lines 2 & 3). If the change in the UP of the goal node in the AG, i.e. ($state_{Goal}$), is less than a predefined threshold, it means that the attack has not a significant effect in compromising the system and hence, we do not perform countermeasure selection. All the descendant nodes of $state_{Compromised}$, including $state_{Compromised}$, are collected into a set $T$ (line 7). After that, for each state in the model, all applicable countermeasures are selected and new probabilities are calculated based on the ability of countermeasures in reducing the probability of exploitations (lines 11 & 12). The influence of each countermeasure is calculated based on the change in the UP of $state_{Goal}$ (line 13). This value represents the total influence of countermeasures in preventing attackers reaching $state_{Goal}$. After calculating influence level of countermeasures on their corresponding states, the utility of them should be calculated based on Equation 2 (line 19). The utility represents the ultimate usefulness and advantageousness of countermeasures based on countermeasures influence on states, suitability of countermeasures, importance level of states and countermeasures intrusiveness. Finally, while there is enough budget, using a greedy method, in each iteration, a countermeasure with the most utility is selected as part of the risk mitigation plan (line 23) and its cost of implementation is subtracted from budget value (line 24).

Algorithm 1. Dynamic Cost-Benefit Analysis

**Input**: IDS alert, AG model, SC Vectors, suitability tables, importance level and budget

**Output**: optimal SRMP

1. Let $state_{Compromised}$ = state node corresponding to the IDS *Alert*
2. Set $Pr(state_{Compromised})$ = 1
3. $UpdateProbabilities()$
4. **if** $(\Delta UP(state_{Goal}) < threshold)$ **then**
5. return
6. end if
7. Let $T = Descendants\ (state_{Compromised}) \cup state_{Compromised}$
8. Let $influence\ [|T|, |SC|] = \emptyset$
9. **for** each $t \in T$ do
10. **for** each $sc \in SC$ do
11. $Pr(t) = Pr(t) \times (1 - sc.probabilityReduction)$
12. $UpdateProbabilities()$
13. $influence\ [t, sc] = \Delta UP(state_{Goal}) \times 100$
14. end for
15. **end for**
16. Let $utility\ [|T|, |SC|] = \emptyset$
17. **for** each $t \in T$ do
18. **for** each $sc \in SC$ do
19. $\text{utility}[t, sc] = \frac{influence[t,sc]*suitability_{sc}*importance_t}{intrusiveness_{sc}}$ (2)
20. **end for**
21. **end for**
22. **while** $(budget > MIN(sc.cost))$
23. $SRMP = SelectOptimalSC\ (utility)$
24. $budget = budget - sc.cost$
25. **end while**
26. **return** $SRMP$

## 5. Experimental Results

In this section, we study a hypothetical network to validate the rationality, feasibility and efficacy of the proposed method.

In the experimental network shown in Figure 3, there are seven hosts, namely, Web server, Mail server, DNS server, Gateway server, SQL server, Administrative server and local desktops which are located within two zones, namely, DMZ zone and Trusted zone. A firewall is used to separate the

DMZ zone (which is accessible to the public) from the trusted zone. Policies allow Web server to send SQL queries to the SQL server. Local desktops and administrative server use remote desktop service which allows remote communication of employees. Moreover, SSHD protocol is installed on the gateway server to monitor remote connections.

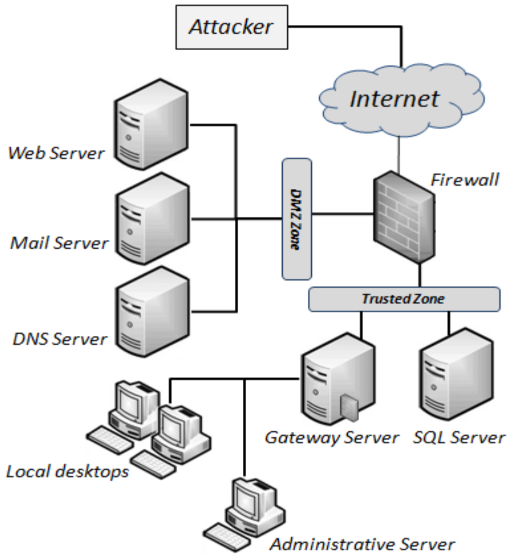



Figure 3. Topology of the Test Network

### *5.1. Vulnerability Scanning*

Having information about the system under assessment and its topology, Security administrator can define an Asset Vector for each asset. These vectors for the test network of Figure 3 are as follow. The definition of Asset Vector is explained in Section 3.

$A_1 = \{Local\ Desktops, Windows\ Server, enabled\ remote\ login, (Partial, Partial, Partial)\}$

$A_2 = \{Administrative\ Server, Windows\ Server, enabled\ remote\ login, (Complete, Complete, Complete)\}$

$A_3 = \{Gateway\ Server, Windows\ Server, Remote\ Desktop\ Serivces, OpenSSH\ 3.7, (None, None, Complete)\}$

$A_4 = \{SQL\ Server, Windows\ Server, Microsoft\ SQL\ Server, (Complete, Complete, Partial)\}$

$A_5 = \{(Mail\ Server, Windows\ Server, Microsoft\ Exchange\ Services, (Complete, None, Partial)\ )\}$

$A_6 = \{DNS\ Server, Windows\ Server, DNS\ protocol, (None, Partial, Complete)\}$

$A_7 = \{Web\ Server, Windows\ Server, Microsoft\ Internet\ Information\ Services\ (IIS), (Partial, Partial, Complete)\}$

Each Asset Vector is comprised of the name of an asset, its running operating system and installed services, software and protocols on it. Moreover, each Asset Vector consists of the values of C, I and A requirements which are assigned by network security administrator. For instance, the availability requirement of asset $A_4$ is *Partial*, while its confidentiality and integrity requirements have *Complete* importance.

After defining Asset Vectors, these assets should be scanned to find their security vulnerabilities. Here, Nessus [51] vulnerability scanner is used. Using the results, we define Vulnerability Vectors as follow. The definition of Vulnerability Vector is explained in Section 3.

$V_1 = \{CVE\ 2019-0708, Remote\ Desktop\ Services, (Complete, Complete, Complete), 1.00\}$

$V_2 = \{CVE\ 2018-17706, Foxit\ PhantomPDF, (Partial, Partial, Partial), 0.68\}$

$V_3 = \{CVE\ 2016-3207, Internet\ Explorer\ 11, (Complete, Complete, Complete), 0.76\}$

$V_4 = \{CVE\ 2017-16381, Adobe\ Acrobat\ and\ Reader, (Complete, Complete, Complete), 0.93\}$

$V_5 = \{CVE\ 2008-0166, OpenSSL\ 0.9.8c-1, (Complete, None, None), 0.78\}$

$V_6 = \{CVE\ 2016-7407, Dropbear\ SSH, (Complete, Complete, Complete), 1.00\}$

$V_7 = \{CVE\ 2007-4752, SSH\ in\ OpenSSH\ before\ 4.7, (Partial, Partial, Partial), 0.7\}$

$V_8 = \{CVE\ 2017-11509, Firebird\ SQL\ Server\ version\ 3.0.2, (Complete, Complete, Complete), 0.90\}$

$V_9 = \{CVE\ 2019-11682, MailCarrier\ 2.51, (Partial, Partial, Partial), 0.75\}$

$V_{10} = \{CVE\ 2008-3060, V-webmail\ 1.5.0, (Partial, None, None), 0.50\}$

$V_{11} = \{CVE\ 2001-1030, Squid\ Proxy\ Server, (Partial, Partial, Partial), 0.75\}$

$V_{12} = \{CVE\ 2010-0290, ISC\ BIND\ 9.7.0, (None, Partial, Partial), 0.40\}$

$V_{13} = \{CVE\ 2017-7269, Internet\ Information\ Services\ (IIS)\ 6.0\ in\ Microsoft\ Windows\ Server\ 2003\ R2, (Complete, Complete, Complete), 1.00\}$

The mapping between corresponding vulnerabilities for each asset is listed in Assets Vulnerabilities table, shown in Table 4.

Table 4. Assets Vulnerabilities Table

| Asset | Vulnerability |
|---|---|
| $A_1$ | $V_1$, $V_2$, $V_3$ |
| $A_2$ | $V_4$ |
| $A_3$ | $V_5$, $V_6$, $V_7$ |
| $A_4$ | $V_8$ |
| $A_5$ | $V_9$, $V_{10}$, $V_{11}$ |
| $A_6$ | $V_{12}$ |
| $A_7$ | $V_{13}$ |

### *5.2. Attack Graph Generation*

Using information about assets vulnerabilities and their connectivity, AG model of the network can be automatically generated using MulVAL network security analyzer [57]. The simple representation of attack graph of the test network is shown in Figure 4.

Attack graph shown in Figure 4 is a directed acyclic graph in which remote attacker's entry point is represented using plain text, security vulnerabilities are represented using ovals and possible attacker's goals are represented using dashed shapes. A directed edge represents a transition from one state to another.

As can be seen, there are several possible goals an attacker can choose to compromise. For simplicity, in this experiment, a scenario of compromising administrative server is considered as the attackers' goal.

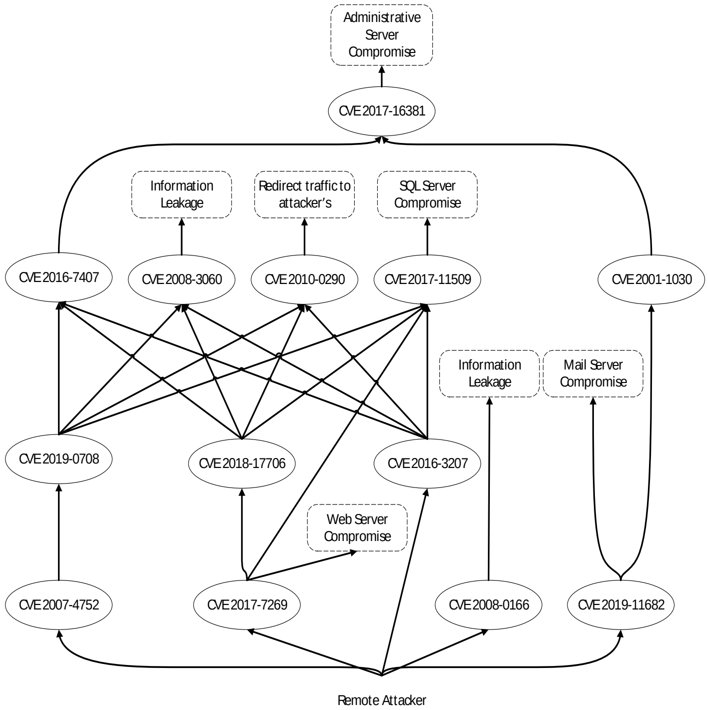



Figure 4. Attack Graph of the Test Network

Table 5. Attack Graph's States Importance

| State ID | State Name | Importance |
|---|---|---|
| S1 | CVE 2007-4752 | 0.1 |
| S2 | CVE 2017-7269 | 0.2 |
| S3 | CVE 2016-3207 | 0.42 |
| S4 | CVE 2008-0166 | NA |
| S5 | CVE 2019-11682 | 0.25 |
| S6 | CVE 2019-0708 | 0.33 |
| S7 | CVE 2018-17706 | 0.33 |
| S8 | CVE 2001-1030 | 0.25 |
| S9 | CVE 2016-7407 | 1.2 |
| S10 | CVE 2008-3060 | NA |
| S11 | CVE 2010-0290 | NA |
| S12 | CVE 2017-11509 | NA |
| S13 | CVE 2017-16381 | 1.5 |

### 5.3. Attack Graph Analysis

In order to identify risk hotspots in the generated AG model, exposure, path length, closeness centrality and betweenness centrality metrics should be extracted from its structure as discussed in Section 4.2.2. After assigning metrics values, AG states importance can be calculated using Equation 1. The results of analyzing test network's AG are represented in Table 5. Detailed information about metrics values are provided in Supplementary Table 1 in Appendix. As can be seen, states $S_{13}$, $S_9$ and $S_3$ have the highest importance among other nodes, respectively.

### 5.4. Bayesian Inference

In order to calculate UP of compromising network states, GeNIe Modeler [62] is used to apply Bayesian inference over the generated AG model. As the result of Bayesian inference, initial UP of compromising network states when no countermeasures is implemented are calculated. These probabilities are listed in second column of Table 6. As can be seen, states $S_9$, $S_2$ and $S_{12}$ have the highest UP among other nodes, meaning that they are the most probable stepping stones for attackers.

Table 6. Unconditional Probabilities of AG States

| State | Initial UP | UP After $S_2$ Compromise |
|---|---|---|
| S1 | 0.528 | 0.747 |
| S2 | 0.696 | 1.00 |
| S3 | 0.535 | 0.757 |
| S4 | 0.549 | 0.777 |
| S5 | 0.528 | 0.747 |
| S6 | 0.527 | 0.742 |
| S7 | 0.476 | 0.680 |
| S8 | 0.401 | 0.563 |
| S9 | 0.702 | 0.978 |
| S10 | 0.531 | 0.744 |
| S11 | 0.461 | 0.646 |
| S12 | 0.687 | 0.959 |
| S13 | 0.405 | 0.570 |

### 5.5. Vulnerability Coverage Assessment

In this process, a Security Countermeasure Vector is defined for each SC covering identified vulnerabilities. List of SCs can be acquired from online repositories, reports, documents and knowledge of security administrator. Vectors of SCs covering the test network vulnerabilities are listed as follow. The definition of Security Countermeasure Vector is explained in Section 3.

$SC_1 = \{Filtering\ external\ traffics, (V_1, V_9, V_{13}),$
$IR: (Partial, Partial, Partial), PR: 0.60, I: Partial, IC: 70\}$

$SC_2 = \{Apply\ MS\ workaround, (V_3),$
$IR: (Complete, Complete, Complete), PR: 0.65,$
$I: None, IC: 30\}$

$SC_3 = \{Disable\ WebDAV, (V_{13}),$
$IR: (Complete, Complete, Complete), PR: 0.95,$
$I: Complete, IC: 120\}$

$SC_4 = \{Patch\ OpenSSH, (V_6, V_7),$
$IR: (Complete, Complete, Complete), PR: 0.75,$
$I: None, IC: 63\}$

$SC_5 = \{Disable\ port\ scan, (V_{11}),$
$IR: (Partial, Partial, Partial), PR: 0.45,$
$I: Partial, IC: 21\}$

$SC_6 = \{Add\ network\ IDS, (V_{10}, V_{11}),$
$IR: (Partial, Partial, Partial), PR: 0.68,$
$I: Partial, IC: 102\}$

$SC_7 = \{Gateway\ firewall, (V_2),$
$IR: (Partial, Partial, Partial), PR: 0.43, I: Partial,$
$IC: 105\}$

$SC_8 = \{Query\ restriction, (V_8),$
$IR: (Partial, Partial, Partial), PR: 0.28, I: Partial, IC: 84\}$

$SC_9 =$
$\{Disable\ external\ UDF\ libraries\ from\ being\ loaded,$
$(V_8), IR: (Partial, Partial, Partial), PR: 0.43, I: Partial,$
$IC: 31\}$

$SC_{10} = \{Upgrade\ firebird\ packages, (V_8),$
$IR: (Partial, Partial, Partial), PR: 0.65, I: None, IC: 45\}$

$SC_{11} = \{Apply\ bind\ security\ update, (V_{12}),$
$IR: (Complete, Complete, Complete), PR: 0.56, I: None,$
$IC: 34\}$

$SC_{12} = \{Limit\ access\ to\ DNS\ server, (V_{12}),$
$IR: (Complete, Complete, Complete), PR: 0.8, I: Partial,$
$IC: 53\}$

$SC_{13} = \{Digital\ signature, (V_{10}),$
$IR: (None, Complete, None), PR: 0.3, I: None, IC: 33\}$

$SC_{14} = \{Use\ POP3, (V_{10}),$
$IR: (None, None, Partial), PR: 0.25, I: None, IC: 153\}$

Having Vulnerability Vectors and Security Countermeasure Vectors, we can assess the efficacy of SCs coverage over vulnerabilities using Coverage Tables. The integrated Coverage Table of countermeasures over their covered vulnerabilities for each one of the C, I and A requirements is shown in Table 7. The detailed Coverage Table containing the values of C, I and A metrics of vulnerabilities and security countermeasures is provided in Supplementary Table 2 in Appendix.

### 5.6. Policy Conformance Assessment

Using Asset Vectors and Coverage Tables created in vulnerability coverage assessment process, Suitability Tables for security countermeasures can be created according to Section 4.3.2. The summarized Suitability Table of countermeasures based on assets policies is shown in Table 8. The detailed Suitability Table for each one of the C, I and A requirements is presented in Supplementary Table 3 in Appendix.

Table 7. Coverage Table for C, I and A Requirements

| Security Countermeasure | Vulnerability | Confidentiality Coverage Level | Integrity Coverage Level | Availability Coverage Level |
|---|---|---|---|---|
| $SC_1$ | $V_1$ | Little coverage | Little coverage | Little coverage |
| | $V_9$ | Equal coverage | Equal coverage | Equal coverage |
| | $V_{13}$ | Little coverage | Little coverage | Little coverage |
| $SC_2$ | $V_3$ | Equal coverage | Equal coverage | Equal coverage |
| $SC_3$ | $V_{13}$ | Equal coverage | Equal coverage | Equal coverage |
| $SC_4$ | $V_6$ | Equal coverage | Equal coverage | Equal coverage |
| | $V_7$ | Extra coverage | Extra coverage | Extra coverage |
| $SC_5$ | $V_{11}$ | Equal coverage | Equal coverage | Equal coverage |
| $SC_6$ | $V_{10}$ | Equal coverage | Extra coverage | Extra coverage |
| | $V_{11}$ | Equal coverage | Equal coverage | Equal coverage |
| $SC_7$ | $V_2$ | Equal coverage | Equal coverage | Equal coverage |
| $SC_8$ | $V_8$ | Little coverage | Little coverage | Little coverage |
| $SC_9$ | $V_8$ | Little coverage | Little coverage | Little coverage |
| $SC_{10}$ | $V_8$ | Little coverage | Little coverage | Little coverage |
| $SC_{11}$ | $V_{12}$ | Extra coverage | Extra coverage | Extra coverage |
| $SC_{12}$ | $V_{12}$ | Extra coverage | Extra coverage | Extra coverage |
| $SC_{13}$ | $V_{10}$ | No coverage | Extra coverage | Equal coverage |
| $SC_{14}$ | $V_{10}$ | No coverage | Equal coverage | Extra coverage |

Table 8. Suitability Values of Security Countermeasures Covering Asset's Vulnerabilities

| Security Countermeasure | Vulnerability | Asset | Total Suitability |
|---|---|---|---|
| $SC_1$ | $V_1$ | $A_1$ | 12 |
| | $V_9$ | $A_5$ | 19 |
| | $V_{13}$ | $A_7$ | 10 |
| $SC_2$ | $V_3$ | $A_1$ | 21 |
| $SC_3$ | $V_{13}$ | $A_7$ | 23 |
| $SC_4$ | $V_6$ | $A_3$ | 15 |
| | $V_7$ | $A_3$ | 12 |
| $SC_5$ | $V_{11}$ | $A_5$ | 19 |
| $SC_6$ | $V_{10}$ | $A_5$ | 17 |
| | $V_{11}$ | $A_5$ | 19 |
| $SC_7$ | $V_2$ | $A_1$ | 21 |
| $SC_8$ | $V_8$ | $A_4$ | 8 |
| $SC_9$ | $V_8$ | $A_4$ | 8 |
| $SC_{10}$ | $V_8$ | $A_4$ | 8 |
| $SC_{11}$ | $V_{12}$ | $A_6$ | 16 |
| $SC_{12}$ | $V_{12}$ | $A_6$ | 16 |
| $SC_{13}$ | $V_{10}$ | $A_5$ | 10 |
| $SC_{14}$ | $V_{10}$ | $A_5$ | 10 |

Table 9. Countermeasures Utility in Preventing $state_{Goal}$ Compromise

| State in T | Covering SC | Influence | suitability | importance | intrusiveness | Utility |
|---|---|---|---|---|---|---|
| S2 | SC1 | 1 | 10 | 0.2 | 0.7 | 2.86 |
| | SC3 | 1.6 | 23 | 0.2 | 1 | 7.36 |
| S7 | SC7 | 0.7 | 21 | 0.33 | 0.7 | 6.93 |
| S9 | SC4 | 1.3 | 15 | 1.2 | 0.3 | 78 |
| S13 | NA | 0.0 | NA | 1.5 | NA | NA |

### *5.7. Dynamic Cost-Benefit Analysis*

Finally, by conducting a dynamic cost-benefit analysis, SRMP with the highest utility and total implementation cost lower than the allocated security hardening budget should be identified. As seen in Section 5.4, in case where no countermeasures is implemented, initial UP of compromising network states is calculated using Bayesian inference technique and the results are listed in the second column of Table 6. It can be seen that the UP of reaching $state_{Goal}$ by attackers, i.e. compromising Administrative Server, is equal to 0.405.

By an assumption that an IDS alert is generated representing state $S_2$ is compromised, the proposed dynamic cost-benefit analysis algorithm (algorithm 1) is employed to find the optimal SRMP. In this case, first of all, the probability of state $S_2$ is changed to 1 (line 2) and the UP of graph states are updated by applying Bayesian inference algorithm (line 3). The result is shown in the third column of Table 6.

Since $\Delta UP(state_{Goal})$ is equal to 0.570 - 0.405 = 0.165 and is more than the predefined threshold, let's say 0.1, the procedure of countermeasure selection continues. In the next step, the set $T = \{S_2, S_7, S_9, S_{13}\}$ is created according to line 7. In lines 9 to 15, the influence of countermeasures in preventing attackers reaching $state_{Goal}$ is calculated.

In lines 17 to 21, the utility of countermeasures is calculated using Equation 2. SCs utility values and metrics values used to calculate them are shown in Table 9.

Finally, assuming that the allocated budget for system hardening is equal to 200 units, the output of the algorithm, according to the utility values of countermeasures, is $SRMP = \{SC_3, SC_4\}$ with the total implementation cost of 183 units.

## 6. Conclusions and Future Works

By increasing the volume and sophistication of today's cyber-attacks, the need for a method to identify the optimal set of security countermeasures is indispensable. This paper presents a dynamic security risk management framework to identify the optimal security risk mitigation plans considering vulnerabilities characteristics, countermeasures effectiveness, existing security policies and budget limitations. Systems risk hotspots are identified by conducting structural analysis of attack graph. By conducting probabilistic analysis of attack graph, the most probable stepping stones for attackers are determined. Countermeasures suitability are calculated according to their ability in covering vulnerabilities and assets security policies. Moreover, a dynamic cost-benefit analysis algorithm is proposed to identify the optimal security risk mitigation plans. Finally, the feasibility and applicability of the proposed framework is ensured using a case study. In future, we try to further extend the proposed framework by considering attackers' capabilities and intentions in bypassing countermeasures and exploiting vulnerabilities.

## 7. References


[1] Ross, R., "Guide for conducting risk assessments NIST special publication 800-30 revision 1", US Dept. Commerce, NIST, Gaithersburg, MD, USA, Tech. Rep, 2012.

[2] Wheeler, E., Security risk management: Building an information security risk management program from the Ground Up. Elsevier, 2011.

[3] Kuzminykh, I., Ghita, B., Sokolov, V., and Bakhshi, T., "Information security risk assessment", *Encyclopedia*, Vol. 1, No. 3, pp. 602–617, 2021.

[4] Shameli-Sendi, A., Cheriet, M., and Hamou-Lhadj, A., "Taxonomy of intrusion risk assessment and response system", *Computers & Security*, Vol. 45, pp. 1–16, 2014.

[5] Shameli-Sendi, A., Aghababaei-Barzegar, R., and Cheriet, M., "Taxonomy of information security risk assessment (ISRA)", *Computers & security*, Vol. 57, pp. 14–30, 2016.

[6] Erdogan, G., and Refsdal, A., "A method for developing qualitative security risk assessment algorithms", *in International Conference on Risks and Security of Internet and Systems*, pp. 244–259, Springer, 2017.

[7] Dobaj, J., Schmittner, C., Krisper, M., and Macher, G., "Towards integrated quantitative security and safety risk assessment", *in International Conference on Computer Safety*, Reliability, and Security, pp. 102–116, Springer, 2019.

[8] Khosravi-Farmad, M., Rezaee, R., Harati, A., and Bafghi, A. G., "Network security risk mitigation using Bayesian decision networks", *in 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 267–272, IEEE, 2014.

[9] Wang, J., Neil, M., and Fenton, N., "A bayesian network approach for cybersecurity risk assessment implementing and extending the fair model", *Computers & Security*, Vol. 89, pp. 101659, 2020.

[10] Hulitt, E., and Vaughn, R. B., "Information system security compliance to FISMA standard: a quantitative measure", *Telecommunication Systems*, Vol. 45, No. 2, pp. 139–152, 2010.

[11] Lo, C.-C., and Chen, W.-J., "A hybrid information security risk assessment procedure considering interdependences between controls", *Expert Systems with Applications*, Vol. 39, No. 1, pp. 247–257, 2012.

[12] Figueira, P. T., Bravo, C. L., and López, J. L. R., "Improving information security risk analysis by including threat-occurrence predictive models", *Computers & Security*, Vol. 88, pp. 101609, 2020.

[13] CVSS, "Common vulnerability scoring system v3.0: Specification document".

[14] FIRST, "Forum of incident response and security teams". https://www.first.org/.

[15] Khosravi-Farmad, M., Ramaki, A. A., and Bafghi, A. G., "Moving target defense against advanced persistent threats for cybersecurity enhancement", *in 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 280–285, IEEE, 2018.

[16] Ouassini, A., and Hunter, M., "Advanced Persistent Threats (APTs)", *The Handbook of Homeland Security*, CRC Press, pp. 163–165, 2023.

[17] Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., and Djukic, P., "Machine learning-enabled IoT security: Open issues and challenges under advanced persistent threats", *ACM Computing Surveys*, Vol. 55, No. 5, pp. 1–37, 2022.

[18] Hong, J. B., Kim, D. S., Chung, C.-J., and Huang, D.,

"A survey on the usability and practical applications of graphical security models", *Computer Science Review*, Vol. 26, pp. 1–16, 2017.

[19] Kaynar, K., "A taxonomy for attack graph generation and usage in network security", *Journal of Information Security and Applications*, Vol. 29, pp. 27–56, 2016.

[20] Lallie, H. S., Debattista, K., and Bal, J., "A review of attack graph and attack tree visual syntax in cyber security", *Computer Science Review*, Vol. 35, pp. 100219, 2020.

[21] Shameli-Sendi, A., and Dagenais, M., "Arito: Cyber-attack response system using accurate risk impact tolerance", *International journal of information security*, Vol. 13, No. 4, pp. 367–390, 2014.

[22] Zahid, M., Inayat, I., Daneva, M., and Mehmood, Z., "A security risk mitigation framework for cyber physical systems", *Journal of software: Evolution and Process*, Vol. 32, No. 2, pp. e2219, 2020.

[23] Li, S., Tryfonas, T., Russell, G., and Andriotis, P., "Risk assessment for mobile systems through a multilayered hierarchical bayesian network", *IEEE transactions on cybernetics*, Vol. 46, No. 8, pp. 1749–1759, 2016.

[24] Shameli-Sendi, A., Louafi, H., He, W., and Cheriet, M., "Dynamic optimal countermeasure selection for intrusion response system", *IEEE Transactions on Dependable and Secure Computing*, Vol. 15, No. 5, pp. 755–770, 2016.

[25] Li, S., Zhao, S., Yuan, Y., Sun, Q., and Zhang, K., "Dynamic security risk evaluation via hybrid bayesian risk graph in cyber-physical social systems", *IEEE Transactions on Computational Social Systems*, Vol. 5, No. 4, pp. 1133–1141, 2018.

[26] He, W., Li, H., and Li, J., "Unknown vulnerability risk assessment based on directed graph models: a survey", *IEEE Access*, Vol. 7, pp. 168201–168225, 2019.

[27] Garg, U., Sikka, G., and Awasthi, L. K., "Empirical analysis of attack graphs for mitigating critical paths and vulnerabilities", *Computers & Security*, Vol. 77, pp. 349–359, 2018.

[28] Hermanowski, D., and Piotrowski, R., "Network risk assessment based on attack graphs", *in International Conference on Dependability and Complex Systems*, pp. 156–167, Springer, 2021.

[29] Rezaee, R., and Ghaemi Bafghi, A., "A risk estimation framework for security threats in computer networks", *Journal of Computing and Security*, Vol. 7, No. 1, pp. 19–33, 2020.

[30] Rezaee, R., Bafghi, A. G., and Khosravi-Farmad, M., "A threat risk estimation model for computer network security", *in 2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 223–228, IEEE, 2016.

[31] Presekal, A., Ştefanov, A., Rajkumar, V. S., and Palensky, P., "Attack graph model for cyber-physical power systems using hybrid deep learning", IEEE Transactions on Smart Grid, 2023.

[32] Liu, Y., and Man, H., "Network vulnerability assessment using bayesian networks", *in Data mining, intrusion detection, information assurance, and data networks security 2005*, Vol. 5812, pp. 61–71, International Society for Optics and Photonics, 2005.

[33] Frigault, M., and Wang, L., "Measuring network security using bayesian network-based attack graphs", *in 2008 32nd Annual IEEE International Computer Software and Applications Conference*, pp. 698–703, IEEE, 2008.

[34] Poolsappasit, N., Dewri, R., and Ray, I., "Dynamic security risk management using bayesian attack graphs", *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 1, pp. 61–74, 2011.

[35] Feng, N., Wang, H. J., and Li, M., "A security risk analysis model for information systems: Causal relationships of risk factors and vulnerability propagation analysis", *Information sciences*, Vol. 256, pp. 57–73, 2014.

[36] Le, A., Chen, Y., Chai, K. K., Vasenev, A., and Montoya, L., "Incorporating fair into bayesian network for numerical assessment of loss event frequencies of smart grid cyber threats", *Mobile Networks and Applications*, Vol. 24, No. 5, pp. 1713–1721, 2019.

[37] Al-Hadhrami, N., Collinson, M., and Oren, N., "A subjective network approach for cybersecurity risk assessment", *in 13th International Conference on Security of Information and Networks*, pp. 1–8, 2020.

[38] Ramaki, A. A., Khosravi-Farmad, M., and Bafghi, A. G., "Real time alert correlation and prediction using Bayesian networks", *in 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 98–103, IEEE, 2015.

[39] Chen, Y. Y., Xu, B., and Long, B., "Information security assessment of wireless sensor networks based on bayesian attack graphs", *Journal of Intelligent & Fuzzy Systems*, Vol. 41, No. 3, pp. 4511–4517, 2021.

[40] Meyur, R., "A bayesian attack tree based approach to assess cyber-physical security of power system", *in 2020 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, IEEE, 2020.

[41] Khosravi-Farmad, M., Ramaki, A. A., and Bafghi, A. G., "Risk-based intrusion response management in ids using bayesian decision networks", *in 2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 307–312, IEEE, 2015.

[42] Behbehani, D., Komninos, N., Al-Begain, K., and Rajarajan, M., "Cloud enterprise dynamic risk assessment (CEDRA): a dynamic risk assessment using dynamic bayesian networks for cloud environment", *Journal of Cloud Computing*, Vol. 12, No. 1, 2023.

[43] Nespoli, P., Papamartzivanos, D., Mármol, F. G., and Kambourakis, G., "Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks", *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 2, pp. 1361–1396, 2017.

[44] Noel, S., Jajodia, S., O'Berry, B., and Jacobs, M., "Efficient minimum-cost network hardening via exploit dependency graphs", *in 19th Annual Computer Security Applications Conference*, 2003. Proceedings., pp. 86–95, IEEE, 2003.

[45] Jha, S., Sheyner, O., and Wing, J., "Two formal analyses of attack graphs", *in Proceedings 15th IEEE Computer Security Foundations Workshop.* CSFW-15,

pp. 49–63, IEEE, 2002.
[46] Dewri, R., Poolsappasit, N., Ray, I., and Whitley, D., "Optimal security hardening using multi-objective optimization on attack tree models of networks", *in Proceedings of the 14th ACM conference on Computer and communications security*, pp. 204–213, 2007.
[47] Khosravi-Farmad, M., and Ghaemi-Bafghi, A., "Bayesian decision network-based security risk management framework", *Journal of Network and Systems Management*, Vol. 28, No. 4, pp. 1794–1819, 2020.
[48] Chung, C.-J., Khatkar, P., Xing, T., Lee, J., and Huang, D., "Nice: Network intrusion detection and countermeasure selection in virtual network systems", *IEEE transactions on dependable and secure computing*, Vol. 10, No. 4, pp. 198–211, 2013.
[49] Schilling, A., and Werners, B., "Optimal selection of it security safeguards from an existing knowledge base", *European Journal of Operational Research*, Vol. 248, No. 1, pp. 318–327, 2016.
[50] Kotenko, I., and Doynikova, E., "Selection of countermeasures against network attacks based on dynamical calculation of security metrics", *The Journal of Defense Modeling and Simulation*, Vol. 15, No. 2, pp. 181–204, 2018.
[51] Nessus, "Nessus vulnerability scanner", Available on, https://www.tenable.com/products/nessus.
[52] OpenVAS, "Open vulnerability assessment scanner", Available on, http://www.openvas.org/.
[53] Retina, "Retina network security vulnerability scanner", Available on, https://www.beyondtrust.com/products/retinanetwork-security-scanner/.
[54] NVD, "NIST US national vulnerability database (NVD)", Available on, https://nvd.nist.gov/.
[55] CVE, "Common vulnerabilities and exposures (CVE)", Available on, https://cve.mitre.org/.
[56] Nmap, "Nmap, the network mapper", Available on, https://nmap.org/.
[57] Ou, X., Govindavajhala, S., Appel, A. W., et al., "Mulval: A logic-based network security analyzer", i*n USENIX security symposium*, Vol. 8, pp. 113–128, Baltimore, MD, 2005.
[58] Jajodia, S., and Noel, S., "Topological vulnerability analysis", *in Cyber situational awareness*, pp. 139–154, Springer, 2010.
[59] Russell, S., and Norvig, P., "Artificial intelligence: A modern approach, global edition 4th", Foundations, Vol. 19, pp. 23, 2021.
[60] Khosravi-Farmad, M., Rezaee, R., and Bafghi, A. G., "Considering temporal and environmental characteristics of vulnerabilities in network security risk assessment", *in 2014 11th International ISC Conference on Information Security and Cryptology*, pp. 186–191, IEEE, 2014.
[61] Koller, D., and Friedman, N., Probabilistic graphical models: principles and techniques. MIT press, 2009.
[62] GeNIe, "GeNIe modeler, bayesfusion, llc", Available on, https://www.bayesfusion.com/.

## 8. Appendix

Supplementary Table 1 represents metrics used for states importance calculation. Detailed coverage table for C, I and A requirements is presented by Supplementary Table 2. Supplementary Table 3 represents the Suitability Table metrics for C, I and A requirements.

Supplementary Table 1. Metrics for Calculating States Importance

| State | Exposure (E) | Path Length (PL) | Closeness Centrality (CC) | Betweenness Centrality (BC) | Importance (I = E×BC / PL×CC) |
|---|---|---|---|---|---|
| S1 | 2 | 5 | 4 | 1 | 0.1 |
| S2 | 4 | 5 | 4 | 1 | 0.2 |
| S3 | 5 | 4 | 3 | 1 | 0.42 |
| S4 | 2 | NA | NA | NA | NA |
| S5 | 3 | 4 | 3 | 1 | 0.25 |
| S6 | 5 | 5 | 3 | 1 | 0.33 |
| S7 | 5 | 5 | 3 | 1 | 0.33 |
| S8 | 2 | 4 | 2 | 1 | 0.25 |
| S9 | 4 | 5 | 2 | 3 | 1.2 |
| S10 | 4 | NA | NA | NA | NA |
| S11 | 4 | NA | NA | NA | NA |
| S12 | 5 | NA | NA | NA | NA |
| S13 | 3 | 4 | 1 | 2 | 1.5 |

Supplementary Table 2. Detailed Coverage Table for C, I and A Requirements

| Security Countermeasure | | | | Vulnerability | | | | Coverage Level | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ID | C | I | A | ID | C | I | A | C | I | A |
| $SC_1$ | P | P | P | $V_1$ | C | C | C | Little | Little | Little |
| | | | | $V_9$ | P | P | P | Equal | Equal | Equal |
| | | | | $V_{13}$ | C | C | C | Little | Little | Little |
| $SC_2$ | C | C | C | $V_3$ | C | C | C | Equal | Equal | Equal |
| $SC_3$ | C | C | C | $V_{13}$ | C | C | C | Equal | Equal | Equal |
| $SC_4$ | C | C | C | $V_6$ | C | C | C | Equal | Equal | Equal |
| | | | | $V_7$ | P | P | P | Extra | Extra | Extra |
| $SC_5$ | P | P | P | $V_{11}$ | P | P | P | Equal | Equal | Equal |
| $SC_6$ | P | P | P | $V_{10}$ | P | N | N | Equal | Extra | Extra |
| | | | | $V_{11}$ | P | P | P | Equal | Equal | Equal |
| $SC_7$ | P | P | P | $V_2$ | P | P | P | Equal | Equal | Equal |
| $SC_8$ | P | P | P | $V_8$ | C | C | C | Little | Little | Little |
| $SC_9$ | P | P | P | $V_8$ | C | C | C | Little | Little | Little |
| $SC_{10}$ | P | P | P | $V_8$ | C | C | C | Little | Little | Little |
| $SC_{11}$ | C | C | C | $V_{12}$ | N | P | P | Extra | Extra | Extra |
| $SC_{12}$ | C | C | C | $V_{12}$ | N | P | P | Extra | Extra | Extra |
| $SC_{13}$ | N | C | N | $V_{10}$ | P | N | N | No | Extra | Equal |
| $SC_{14}$ | N | N | P | $V_{10}$ | P | N | N | No | Equal | Extra |

Supplementary Table 3. Suitability Table Metrics for C, I and A Requirements

| Security Countermeasure | Vulnerability | Coverage Level | | | Asset Policies | | | | Suitability | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | ID | C | I | A | ID | C | I | A | C | I | A | Total |
| $SC_1$ | $V_1$ | Little | Little | Little | $A_1$ | P | P | P | 4 | 4 | 4 | 12 |
| | $V_9$ | Equal | Equal | Equal | $A_5$ | C | N | P | 9 | 3 | 7 | 19 |
| | $V_{13}$ | Little | Little | Little | $A_7$ | P | P | C | 4 | 4 | 2 | 10 |
| $SC_2$ | $V_3$ | Equal | Equal | Equal | $A_1$ | P | P | P | 7 | 7 | 7 | 21 |
| $SC_3$ | $V_{13}$ | Equal | Equal | Equal | $A_7$ | P | P | C | 7 | 7 | 9 | 23 |
| $SC_4$ | $V_6$ | Equal | Equal | Equal | $A_3$ | N | N | C | 3 | 3 | 9 | 15 |
| | $V_7$ | Extra | Extra | Extra | $A_3$ | N | N | C | 2 | 2 | 8 | 12 |
| $SC_5$ | $V_{11}$ | Equal | Equal | Equal | $A_5$ | C | N | P | 9 | 3 | 7 | 19 |
| $SC_6$ | $V_{10}$ | Equal | Extra | Extra | $A_5$ | C | N | P | 9 | 2 | 6 | 17 |
| | $V_{11}$ | Equal | Equal | Equal | $A_5$ | C | N | P | 9 | 3 | 7 | 19 |
| $SC_7$ | $V_2$ | Equal | Equal | Equal | $A_1$ | P | P | P | 7 | 7 | 7 | 21 |
| $SC_8$ | $V_8$ | Little | Little | Little | $A_4$ | C | C | P | 2 | 2 | 4 | 8 |
| $SC_9$ | $V_8$ | Little | Little | Little | $A_4$ | C | C | P | 2 | 2 | 4 | 8 |
| $SC_{10}$ | $V_8$ | Little | Little | Little | $A_4$ | C | C | P | 2 | 2 | 4 | 8 |
| $SC_{11}$ | $V_{12}$ | Extra | Extra | Extra | $A_6$ | N | P | C | 2 | 6 | 8 | 16 |
| $SC_{12}$ | $V_{12}$ | Extra | Extra | Extra | $A_6$ | N | P | C | 2 | 6 | 8 | 16 |
| $SC_{13}$ | $V_{10}$ | No | Extra | Equal | $A_5$ | C | N | P | 1 | 2 | 7 | 10 |
| $SC_{14}$ | $V_{10}$ | No | Equal | Extra | $A_5$ | C | N | P | 1 | 3 | 6 | 10 |